F　プログラム

F.1　マシン・イプシロンを計算する

```
     program epsilon
     real * 8 a,b,c,eps
        a = 4.0d0/3.0d0
        b = a - 1.0d0
        c = b + b + b
        eps = dabs(c - 1.d0)
        write(*, *) eps
        stop
     end
```

F.2　ルジャンドル多項式の 0 点を求める

```
      program legtst
      implicit real*8(a-h,o-z)
      dimension xp(0:200),xq(0:200)

        m=40
        call leg(m,xp,xq)
        sss=0.d0
        do 10 k=1,m-1
          sss=sss+xq(k)*dsqrt(1.d0-xp(k)**2)
   10   continue
        write(6,*) ’2*sss = ’,2*sss
        stop
      end
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
      subroutine leg(m,xp,xq)
```

```
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c    this subroutine gives the collocation points and coefficients
c    in gauss-legendre integral method by bisection method
c
c    p(n)(x) = 0,  (x = x1, x2, x3, ...., xn)
c
c
c    evem function  n = 2*mm
c
c    -1                   0                    +1
c     ------------------------------------------
c     0  1            mm     mm+1         2*mm 2*mm+1
c        *            *
c                2*mm collocation points
c
c    odd function   n = 2*mm-1
c
c    -1                   0                    +1
c     ------------------------------------------
c     0  1               mm            2*mm-1  2*mm
c        *                *
c                2*mm-1 collocation points
c
c
c       +1
c     integ f(x) = sigma xq(k) * f(xp(k))
c       -1          k=1,n
c
      implicit real*8(a-h,o-z)
      dimension xp(0:m),xq(0:m),p(3),q(3),r(3)
        n=m-1
        pi=4*datan(1.d0)
```

```
eps=1.d-14
do 50 k=1,n
  kp1=k+1
  km1=k-1
  if(n/2*2.eq.n) then
    mm=n/2
    if(k.lt.mm) then
      xn=dble(n+1-2*k)/dble(2*n+1)
      xnp1=dble(n+1-2*kp1)/dble(2*n+1)
    else if(k.eq.mm) then
      xn=dble(n+1-2*k)/dble(2*n+1)
      xnp1=0.d0
    else if(k.eq.mm+1) then
      xn=0.d0
      xnp1=dble(n+1-2*k)/dble(2*n+1)
    else
      xn=dble(n+1-2*km1)/dble(2*n+1)
      xnp1=dble(n+1-2*k)/dble(2*n+1)
    endif
  else
    mm=(n+1)/2
    if(k.lt.mm-1) then
      xn=dble(n+1-2*k)/dble(2*n+1)
      xnp1=dble(n+1-2*kp1)/dble(2*n+1)
    else if(k.eq.mm-1) then
      xn=dble(n+1-2*k)/dble(2*n+1)
      xnp1=-eps
    else if(k.eq.mm) then
      xn=0.d0
      xnp1=0.d0
    else if(k.eq.mm+1) then
      xn=eps
      xnp1=dble(n+1-2*k)/dble(2*n+1)
    else
```

```
              xn=dble(n+1-2*km1)/dble(2*n+1)
              xnp1=dble(n+1-2*k)/dble(2*n+1)
            endif
          endif
c         xn=dble(n+1-2*k)/dble(2*n+1)
c         xnp1=dble(n+1-2*kp1)/dble(2*n+1)
          x=dsin(xn*pi)
          xp1=dsin(xnp1*pi)
          write(6,*) ' k = ',k,' x = ',x,'  xp1 = ',xp1
   15     xmd=(x+xp1)/2.d0
c         write(6,*) ' x = ',x,'  xp1 = ',xp1
          if(dabs(x-xp1).lt.eps) go to 30
   10     p(1)=1.d0
          p(2)=x
          q(1)=1.d0
          q(2)=xp1
          r(1)=1.d0
          r(2)=xmd
          do 20 i=2,n
            p(3)=x*p(2)+dble(i-1)/dble(i)*(x*p(2)-p(1))
            p(1)=p(2)
            p(2)=p(3)
            q(3)=xp1*q(2)+dble(i-1)/dble(i)*(xp1*q(2)-q(1))
            q(1)=q(2)
            q(2)=q(3)
            r(3)=xmd*r(2)+dble(i-1)/dble(i)*(xmd*r(2)-r(1))
            r(1)=r(2)
            r(2)=r(3)
   20     continue
          fac1=p(2)*r(2)
          fac2=q(2)*r(2)
          if(fac1.lt.0.d0) then
            xp1=xmd
            go to 15
```

```
    endif
    if(fac2.lt.0.d0) then
      x=xmd
      go to 15
    endif
 30 continue
 40 w=2.d0*(1.d0-x**2)/(dble(n)*p(1))**2
    y=-x
    xp(k)=y
    xq(k)=w
 50 continue
   write(6,650) (k,xp(k),xq(k),k=1,m-1)
650 format(2x,'k=',i3,6x,'y=',d14.7,6x,'w=',d14.7)
   return
  end
```

F.3　2 重指数型積分公式

```
c
c    program for integration
c    by Double Exponential Type Integration Method
c
c    example $\int_{-1}^{1} \sin\left(\frac{\pi}{2}(x+1)\right) dx = \frac{4}{\pi}$
c
c    gives the values for rl and ne
c
  program double
  implicit real*8(a-h, o-z)
  parameter(nn=10000)
  dimension fac(0:nn), fp(0:nn), xp(0:nn), xq(0:nn)
  data ne, rl, a, b/100, 10.d0, -1.d0, 1.d0/
  pi=dacos(-1.d0)
  h=2.d0*rl/dble(ne)
  do 70 j=0, ne
```

```
      fac(j)=1.d0
      if(j.eq.0.or.j.eq.ne) then
      fac(j)=0.5d0
      endif
      xp(j)=-rl+h*dble(j)
      xpp=xp(j)
      yy=dtanh(pi/2.d0*dsinh(xpp))
      xq(j)=(b-a)/2.d0*yy+(a+b)/2.d0
      if(xpp.ge.0.d0) then
      fp(j)=2.d0*(1.d0+dexp(-2.d0*xpp))*dexp(-pi*dsinh(xpp))
     &       /(dexp(-xpp)+2.d0*dexp(-xpp-pi*dsinh(xpp))
     &       +dexp(-xpp-2.d0*pi*dsinh(xpp)))
      else
      fp(j)=2.d0*(1.d0+dexp(2.d0*xpp))*dexp(pi*dsinh(xpp))
     &       /(dexp(xpp)+2.d0*dexp(xpp+pi*dsinh(xpp))
     &       +dexp(xpp+2.d0*pi*dsinh(xpp)))
      endif
   70 continue
      sss=0.d0
      do 10 j=0, ne
      xqq=xq(j)
      call func(xqq,ff)
      sss=sss+fac(j)*ff*fp(j)
   10 continue
      sss=sss*h*pi/4.d0*(b-a)
      write(6,100) ne,sss
  100 format(2x, ' ne = ', i5, ' sss = ', 1pe16.9)
      end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
      subroutine func(x, f)
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      implicit real*8(a-h, o-z)
      pi=dacos(-1.d0)
      f=dsin(pi/2.d0*(x+1.d0))
      return
   end
```